

Bisimilarity on Basic Process Algebra is in 2-ExpTime (an explicit proof)

Petr Jančar

Techn. Univ. Ostrava, Czech Republic, <http://www.cs.vsb.cz/jancar/>

Abstract

Burkart, Caucal, Steffen (1995) showed a procedure deciding bisimulation equivalence of processes in Basic Process Algebra (BPA), i.e. of sequential processes generated by context-free grammars. They improved the previous decidability result of Christensen, Hüttel, Stirling (1992), since their procedure has obviously an elementary time complexity and the authors claim that a close analysis would reveal a double exponential upper bound. Here a self-contained direct proof of the membership in 2-ExpTime is provided. This is done via a Prover-Refuter game which shows that there is an alternating Turing machine deciding the problem in exponential space. The proof uses similar ingredients (size-measures, decompositions, bases) as the previous proofs, but one new simplifying factor is an explicit addition of infinite regular strings to the state space. To have an explicit proof of the upper bound seems useful also due to the recent shift of the known lower bound from PSPACE (Srba, 2002) to ExpTime (Kiefer, 2012).

1 Introduction

The classical language equivalence problems in automata theory have their counterparts in the bisimulation equivalence problems in process theory. The computational complexity of bisimulation equivalence is still not fully settled even for fundamental classes, one of them being the class of Basic Process Algebra (BPA) processes, i.e. of sequential processes generated by context-free grammars. This concrete research topic started with a result by Baeten, Bergstra, Klop [1] who showed decidability in the normed BPA case (where each nonterminal of the underlying context-free grammar derives some terminal word). Christensen, Hüttel, Stirling [5] extended the decidability result to the whole BPA class, and Burkart, Caucal, Steffen [4] (see also [3]) showed a procedure with an elementary complexity, claiming that a close analysis would demonstrate a double exponential upper bound. We also note that the normed case was subsequently shown to be in PTIME [7] (see [6] for the most recent improvement of complexity).

Regarding the lower bounds for the (full) BPA problem, Srba [15] showed PSPACE-hardness, and Kiefer [11] recently shifted this to ExpTime-hardness (using the ExpTime-completeness of countdown games [10]); he thus also strengthened the lower bound results known for (visibly) pushdown processes [12], [16] and for weak bisimilarity [13]. This was a bit surprising since the bisimulation problem for related classes of basic parallel processes (generated by commutative context-free grammars) and of one-counter processes were shown PSPACE-complete [8], [2]. The mentioned shift of the lower bound is a natural impulse for

looking at the complexity again, and confirming the upper bound which has been a bit vaguely stated in the literature becomes more important.

Here we show a direct self-contained proof of the fact that BPA bisimilarity is indeed in 2-ExpTime. This is done via a Prover-Refuter game which shows that there is an alternating Turing machine deciding the problem in exponential space. The proof uses similar ingredients (size-measures, decompositions, bases) as the previous proofs, though in somewhat different form, while the main new simplifying factor seems to be an explicit addition of infinite regular strings to the state space. On the whole, the proof confirms the previously claimed upper bound, simplifies several technical aspects, and it might also shed some new light on the structural decomposition approach for deciding bisimilarity.

Section 2 recalls the notion of regular strings, defines the bisimilarity problem for BPA and states the result. Section 3 then shows a proof. It first recalls some simple notions and observations, noting the congruence properties and defining decompositions, and then a Prover-Refuter game is formulated whose soundness is obvious. Regarding the completeness, the main technical ingredient is, in fact, an exponential bound on the “equivalence-level” of any nonbisimilar normed pair; this can be derived from the literature but, to be self-contained, Subsection 3.1 shows a proof via an explicit presentation of Attacker’s winning strategies in the bisimulation game. Section 4 adds some further remarks.

2 Preliminaries

Let $\mathbb{N} = \{0, 1, 2, \dots\}$. For a (finite) set \mathcal{C} , $|\mathcal{C}|$ denotes its cardinality, and \mathcal{C}^* the set of finite sequences (strings, words) of elements of \mathcal{C} . By ε we denote the empty sequence and by $|w|$ the length of $w \in \mathcal{C}^*$. \mathcal{C}^ω is the set of infinite strings over \mathcal{C} , i.e. the set of mappings $\mathbb{N} \rightarrow \mathcal{C}$. If $w = uv$ then u is a *prefix* of w and v a *suffix* of w .

Regular strings

A *regular string* over \mathcal{C} is either a finite string (an element of \mathcal{C}^*) or an infinite string (an element of \mathcal{C}^ω) of the form $\beta\gamma\gamma\gamma\dots = \beta\gamma^\omega$ where $\beta, \gamma \in \mathcal{C}^*$ and $\gamma \neq \varepsilon$. We do not consider nonregular strings. For $\alpha \in \mathcal{C}^*$ we put $\text{ROUND}(\alpha) = \{\gamma\beta \mid \beta\gamma = \alpha\}$, and we recall some standard and/or easy facts, which imply a canonical form of regular strings; we stipulate $\varepsilon^\omega = \varepsilon$.

Proposition 1 *If $\beta_1(\gamma_1)^\omega = \beta_2(\gamma_2)^\omega$ then $(\gamma_2)^\omega = (\gamma'_1)^\omega$ for some $\gamma'_1 \in \text{ROUND}(\gamma_1)$.*

Proof: Since $\beta_1\gamma_1\gamma_1\gamma_1\dots = \beta_2\gamma_2\gamma_2\gamma_2\dots$, we obviously must have $\gamma_2\gamma_2\gamma_2\dots = \delta\gamma_1\gamma_1\gamma_1\dots$ for a suffix δ of γ_1 ; let $\gamma_1 = \delta'\delta$. Hence $(\gamma_2)^\omega = \delta(\delta'\delta)^\omega = (\delta\delta')^\omega$. \square

Proposition 2 *Each regular string α has the unique prefix α_p and the unique cycle α_c such that $\alpha = \alpha_p(\alpha_c)^\omega$ and $\alpha_p\alpha_c$ is the shortest possible (i.e., $\alpha = \beta\gamma^\omega$ implies $|\beta\gamma| \geq |\alpha_p\alpha_c|$).*

Proof: Suppose $\alpha = \beta_1(\gamma_1)^\omega = \beta_2(\gamma_2)^\omega$ where $|\gamma_1| < |\gamma_2|$. Prop. 1 shows that $\alpha = \beta_2(\gamma'_1)^\omega$ for some $\gamma'_1 \in \text{ROUND}(\gamma_1)$, and we have $|\beta_2\gamma'_1| < |\beta_2\gamma_2|$. \square

We call $\alpha_p(\alpha_c)^\omega$ the *canonical presentation* of α (where $\alpha_p = \alpha$ and $\alpha_c = \varepsilon$ when α is finite). The following corollary is useful later; in particular it says that changing a finite prefix does not change the canonical cycle, up to “rounding.”

Proposition 3 (1) If β is finite then α_p is a suffix of $(\beta\alpha)_p$, and $\alpha_c \in \text{ROUND}((\beta\alpha)_c)$.
(2) For any finite β_1, β_2 and any (regular) α we have $(\beta_2\alpha)_c \in \text{ROUND}((\beta_1\alpha)_c)$.

Proof: (1) If $(\beta\alpha)_p$ is a prefix of β then obviously $\alpha_p = \varepsilon$ and $\alpha_c \in \text{ROUND}((\beta\alpha)_c)$; otherwise it is easy to verify that $(\beta\alpha)_p = \beta\alpha_p$ and $\alpha_c = (\beta\alpha)_c$.
(2) follows from (1), when noting that $\alpha \in \text{ROUND}(\gamma)$ implies $\text{ROUND}(\alpha) = \text{ROUND}(\gamma)$. \square

BPA processes

A *BPA-system* is defined as a context-free grammar in Greibach normal form with no starting nonterminal; it is a tuple $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ where $\mathcal{N}, \mathcal{A}, \mathcal{R}$ are finite sets of *nonterminals* (or variables), *actions* (or terminals), and *rewriting rules*, respectively. The rules $r \in \mathcal{R}$ are of the form $r : A \xrightarrow{a} \alpha$ where $A \in \mathcal{N}$, $a \in \mathcal{A}$, $\alpha \in \mathcal{N}^*$; we put $\text{ACT}(r) = a$, thus defining the mapping $\text{ACT} : \mathcal{R} \rightarrow \mathcal{A}$. For later convenience we assume that for each $A \in \mathcal{N}$ there is at least one rule of the form $A \xrightarrow{a} \alpha$, i.e., there are *no dead nonterminals*. (But there may still be nonterminals which do not derive any terminal word in the classical language sense.)

A BPA system $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ generates the (rule-based) labelled transition system $\mathcal{L}_{\mathcal{G}}^{\mathcal{R}} = (\mathcal{S}_{\mathcal{G}}, \mathcal{R}, (\xrightarrow{r})_{r \in \mathcal{R}})$ where $\mathcal{S}_{\mathcal{G}}$ is the set of all *regular* strings over \mathcal{N} , which are also called *states* or *processes*. The *transition relations* $\xrightarrow{r} \subseteq \mathcal{S}_{\mathcal{G}} \times \mathcal{S}_{\mathcal{G}}$ are defined as follows: for a rule $r : A \xrightarrow{a} \alpha$ we have $A\beta \xrightarrow{r} \alpha\beta$ for any regular string β . In the (action-based) labelled transition system $\mathcal{L}_{\mathcal{G}}^{\mathcal{A}} = (\mathcal{S}_{\mathcal{G}}, \mathcal{A}, (\xrightarrow{a})_{a \in \mathcal{A}})$ we have $\alpha \xrightarrow{a} \beta$ if there is r such that $\alpha \xrightarrow{r} \beta$ in $\mathcal{L}_{\mathcal{G}}^{\mathcal{R}}$ and $\text{ACT}(r) = a$. In both $\mathcal{L}_{\mathcal{G}}^{\mathcal{R}}$ and $\mathcal{L}_{\mathcal{G}}^{\mathcal{A}}$ we define \xrightarrow{w} ($w \in \mathcal{R}^*$ or $w \in \mathcal{A}^*$) as usual: $\alpha \xrightarrow{\varepsilon} \alpha$; if $\alpha \xrightarrow{a} \beta$ and $\beta \xrightarrow{u} \gamma$ then $\alpha \xrightarrow{au} \gamma$.

Remark. We note that $\mathcal{L}_{\mathcal{G}}^{\mathcal{R}}$ is deterministic: if $\alpha \xrightarrow{w}$, i.e. if $w \in \mathcal{R}^*$ is enabled by α , then there is a unique path $\alpha \xrightarrow{r_1} \beta_1 \xrightarrow{r_2} \beta_2 \cdots \xrightarrow{r_k} \beta_k$ where $r_1 r_2 \dots r_k = w$. $\mathcal{L}_{\mathcal{G}}^{\mathcal{A}}$ might be nondeterministic. We also note that if α is a finite string and $\alpha \xrightarrow{w} \beta$ then β is also finite. The convenience of including also infinite regular strings into $\mathcal{S}_{\mathcal{G}}$ will become clear later.

Bisimilarity problem for BPA

Given $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, we say that $\mathcal{B} \subseteq \mathcal{S}_{\mathcal{G}} \times \mathcal{S}_{\mathcal{G}}$ *covers* $(\alpha, \beta) \in \mathcal{S}_{\mathcal{G}} \times \mathcal{S}_{\mathcal{G}}$ (in $\mathcal{L}_{\mathcal{G}}^{\mathcal{A}}$) if for any $\alpha \xrightarrow{a} \alpha'$ there is $\beta \xrightarrow{a} \beta'$ such that $(\alpha', \beta') \in \mathcal{B}$, and for any $\beta \xrightarrow{a} \beta'$ there is $\alpha \xrightarrow{a} \alpha'$ such that $(\alpha', \beta') \in \mathcal{B}$. \mathcal{B} *covers* $\mathcal{B}' \subseteq \mathcal{S}_{\mathcal{G}} \times \mathcal{S}_{\mathcal{G}}$ if \mathcal{B} covers each $(\alpha, \beta) \in \mathcal{B}'$. \mathcal{B} is a *bisimulation* if \mathcal{B} covers \mathcal{B} . States α, β are *bisimilar*, $\alpha \sim \beta$, if there is a bisimulation \mathcal{B} containing (α, β) .

The problem BPA-BISIM asks, given \mathcal{G} and two nonterminals X, Y , if $X \sim Y$ (in $\mathcal{L}_{\mathcal{G}}^{\mathcal{A}}$). We will prove the next theorem, assuming a standard encoding of \mathcal{G}, X, Y .

Theorem 4 BPA-BISIM is in 2-EXPTIME; i.e., there is an algorithm which decides BPA-BISIM and its time complexity is in $O(2^{2^{\text{pol}(n)}})$ for a polynomial pol .

3 Proof of Theorem 4

We assume a given $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, generating $\mathcal{L}_{\mathcal{G}}^{\mathcal{A}}$ with the state set $\mathcal{S}_{\mathcal{G}}$, and we recall/define further technical notions. Let $\sim_0 = \mathcal{S}_{\mathcal{G}} \times \mathcal{S}_{\mathcal{G}}$ and let $\sim_{k+1} \subseteq \mathcal{S}_{\mathcal{G}} \times \mathcal{S}_{\mathcal{G}}$ ($k \in \mathbb{N}$) be the set of all pairs covered by \sim_k . The following claims are standard and/or straightforward; for convenience we can write $\alpha\beta$ also if α is infinite but we identify $\alpha\beta$ with α in such a case.

Proposition 5 (1) The relations \sim and \sim_i (for all $i \in \mathbb{N}$) are equivalences.
(2) If $\alpha \sim_{i+1} \beta$ then $\alpha \sim_i \beta$ (hence $\sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \dots$).
(3) We have $\alpha \sim \beta$ iff $\forall i \in \mathbb{N} : \alpha \sim_i \beta$ (since \mathcal{L}_G^A is finitely branching and thus image finite).
(4) If $\alpha \sim_i \beta$ and $\gamma \sim_i \delta$ then $\alpha\gamma \sim_i \beta\delta$. Hence \sim and \sim_i are congruences wrt concatenation.
(5) If $\alpha \sim_i \gamma\alpha$ and $\gamma \neq \varepsilon$ (and $\gamma \in \mathcal{N}^*$) then $\alpha \sim_i \gamma^\omega$.
(6) If $U \in \mathcal{N}$ and there is no $w \in \mathcal{A}^*$ such that $U \xrightarrow{w} \varepsilon$ then $U \sim U\alpha$ for any α .

We note that $\alpha \not\sim_1 \beta$ iff α, β enable different sets of actions. Due to our assumption that there is no dead nonterminal $A \in \mathcal{N}$, we have $\varepsilon \not\sim_1 \alpha$ iff $\alpha \neq \varepsilon$; thus (4) indeed holds.

Remark. The “no dead nonterminal” assumption is not crucial for the problem BPA-BISIM, since we can always add a special nonterminal D and a special action d , and the rules $A \xrightarrow{d} A$ for all dead nonterminals A (including D), and finally replace the question $X \stackrel{?}{\sim} Y$ with $XD \stackrel{?}{\sim} YD$.

Prop. 5 suggests to define the *equivalence level*, or the *eq-level*, for each pair of strings:

$$\text{EqLv}(\alpha, \beta) = k \in \mathbb{N} \text{ if } \alpha \sim_k \beta \text{ and } \alpha \not\sim_{k+1} \beta, \text{ and } \text{EqLv}(\alpha, \beta) = \omega \text{ if } \alpha \sim \beta.$$

Point (6) in Prop. 5 touches upon the notion of *norm* (a mapping $\mathcal{S}_G \rightarrow \mathbb{N} \cup \{\omega\}$), which is the same in both \mathcal{L}_G^R and \mathcal{L}_G^A ; we refer to \mathcal{L}_G^R now:

Definition 6 The norm of $\alpha \in \mathcal{S}_G$ is denoted $\|\alpha\|$: if there is no w such that $\alpha \xrightarrow{w} \varepsilon$ then we put $\|\alpha\| = \omega$ and say that α is *unnormed*; otherwise α is *normed* and $\|\alpha\| = |w|$ for a shortest w such that $\alpha \xrightarrow{w} \varepsilon$. A path $\beta_0 \xrightarrow{r_1} \beta_1 \xrightarrow{r_2} \beta_2 \dots \xrightarrow{r_k} \beta_k$ is *norm-reducing* if $\|\beta_0\| < \omega$ and $\|\beta_{i+1}\| < \|\beta_i\|$ (i.e., $\|\beta_{i+1}\| = \|\beta_i\| - 1$) for $i = 0, 1, \dots, k-1$.

We stipulate $n < \omega$ and $n + \omega = \omega + n = \omega + \omega = \omega$ for each $n \in \mathbb{N}$, and note that $\|\varepsilon\| = 0$ and $\|\alpha\beta\| = \|\alpha\| + \|\beta\|$. (We have $\|\alpha\| = \omega$ when α is infinite.)

Convention. Recalling Point (6) in Prop. 5, we further implicitly assume that each considered string is stripped of the suffix after the first occurrence of an unnormed nonterminal, if any. The considered strings are thus of the form $\alpha, \alpha U, \beta\gamma^\omega$ where α, β, γ are normed and $U \in \mathcal{N}$ is unnormed. We can still write, e.g., $\gamma\beta$ or γ^ω even if $\|\gamma\| = \omega$ but such strings are implicitly identified with (the appropriate prefix of) γ .

It will be useful to “measure” the (norm-)size of string presentations:

Definition 7 Given $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, the function $\text{SIZE} : \mathcal{S}_G \rightarrow \mathbb{N}$ is defined as follows. For a finite string α we put $\text{SIZE}(\alpha) = \|\alpha'\|$ for the longest normed prefix α' of α (thus $\text{SIZE}(\alpha U) = \text{SIZE}(\alpha)$ when U is unnormed). For an infinite regular string α , with no unnormed nonterminal, we put $\text{SIZE}(\alpha) = \|\alpha_p \alpha_c\|$ (where $\alpha_p(\alpha_c)^\omega$ is the canonical presentation of α). For a pair (α, β) we put $\text{SIZE}(\alpha, \beta) = \max\{\text{SIZE}(\alpha), \text{SIZE}(\beta)\}$. We put $\max \emptyset = 0$, and define:

$$\begin{aligned} M &= \max\{\|A\| \mid A \in \mathcal{N}, \|A\| < \omega\}, \\ M_{rhs} &= \max\{\|\alpha\| \mid \text{there is a rule } r : A \xrightarrow{a} \alpha \text{ and } \|\alpha\| < \omega\}, \\ S_{rhs} &= \max\{\text{SIZE}(\alpha) \mid \text{there is a rule } r : A \xrightarrow{a} \alpha\}. \text{ (Hence } M_{rhs} \leq S_{rhs}.) \end{aligned}$$

We note that $\|A\| = 1 + \|\alpha\|$ for *some* rule $A \xrightarrow{a} \alpha$, and we easily observe the following standard facts.

Proposition 8

- (1) There is a polynomial algorithm which, given $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, computes $\|A\|$ for each $A \in \mathcal{N}$, and also M, M_{rhs}, S_{rhs} ; these values are bounded by an exponential function of the size of \mathcal{G} .
- (2) There is a polynomial algorithm which, given β, γ , finds the canonical prefix $(\beta\gamma^\omega)_p$ and cycle $(\beta\gamma^\omega)_c$ and computes $\text{SIZE}(\beta\gamma^\omega)$.

We now define the crucial notion, on which the later Prover-Refuter game is based.

Definition 9 A set of “component-pairs” $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_k, \beta_k)$ is a decomposition of (α, β) if $\text{SIZE}(\alpha_j, \beta_j) < \text{SIZE}(\alpha, \beta)$ for $j = 1, 2, \dots, k$, and (α, β) belongs to the least congruence (wrt concatenation) containing all (α_j, β_j) , $j = 1, 2, \dots, k$. Moreover, if $\alpha_j \sim \beta_j$ for all $j = 1, 2, \dots, k$ then it is a bisimilar decomposition.

Example. One decomposition of $(A\alpha, B\beta)$ is $\{(A\gamma, B), (\alpha, \gamma\beta)\}$ when both $\text{SIZE}(A\gamma, B)$, $\text{SIZE}(\alpha, \gamma\beta)$ are less than $\text{SIZE}(A\alpha, B\beta)$; indeed, the least congruence containing $(A\gamma, B)$, $(\alpha, \gamma\beta)$ must also contain $(A\gamma\beta, B\beta)$ and $(A\alpha, A\gamma\beta)$, and thus also $(A\alpha, B\beta)$. Another decomposition of $(A\alpha, B\beta)$ is $(\alpha, \gamma\delta^\omega), (\beta, \delta^\omega), (A\gamma\delta^\omega, B\delta^\omega)$ if the size conditions are satisfied.

By inspecting the definitions and recalling Prop. 5 (Points (1) and (4)) we easily derive:

Proposition 10

- (1) If $\{(\alpha_j, \beta_j) \mid 1 \leq j \leq k\}$ is a decomposition of (α, β) then $\min \{ \text{EqLv}(\alpha_j, \beta_j) \mid 1 \leq j \leq k \} \leq \text{EqLv}(\alpha, \beta)$; if it is a bisimilar decomposition then $\alpha \sim \beta$.
- (2) If $\text{EqLv}(\alpha, \beta) < \omega$ then there is a move $\alpha \xrightarrow{a} \alpha'$ or $\beta \xrightarrow{a} \beta'$ such that for any $\beta \xrightarrow{a} \beta'$ in the first case and for any $\alpha \xrightarrow{a} \alpha'$ in the second case we have $\text{EqLv}(\alpha', \beta') < \text{EqLv}(\alpha, \beta)$.

We now recall the general fact that $2\text{-ExpTime} = \text{AExpSpace}$ (where “A” stands for “Alternating”); an algorithm proving Theorem 4 can be thus based on the following game. (The algorithm finds if there is a winning strategy of Refuter.)

PROVER (she) - REFUTER (he) GAME

1. A BPA-system $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ and $X, Y \in \mathcal{N}$ are given.
2. A work space of size $2^{\text{pol}(\text{size}(\mathcal{G}))}$ is reserved, where pol is a (sufficient) polynomial whose existence will become clear later. A special part of the work space serves for storing a presentation of a *current pair*, initially (X, Y) .
3. For $i = 1, 2, \dots$, the following Phase i is performed; (α, β) denotes the current pair:
 - (a) If $\alpha \not\sim_1 \beta$ then Refuter wins. If α, β are dead (i.e., if they do not enable any action, i.e. $\alpha = \beta = \varepsilon$) then Prover wins. (The play finishes in these cases.)
 - (b) Using the free work space, Prover can show some pairs and demonstrate that they constitute a decomposition of (α, β) . In this case Refuter chooses one of these pairs as the new current pair (α', β') ; the play then continues with Phase $i+1$.
 - (c) (Prover has not used the possibility in (b).) Refuter chooses a move $\alpha \xrightarrow{a} \alpha'$ or $\beta \xrightarrow{a} \beta'$. In the first case Prover chooses some $\beta \xrightarrow{a} \beta'$, in the second case Prover chooses some $\alpha \xrightarrow{a} \alpha'$. Then (α', β') becomes the new current pair. If (α', β') does not fit into the space reserved for the current pair then Refuter wins; otherwise the play continues with Phase $i+1$.

Lemma 11 (*Soundness.*) *If $X \not\sim Y$ then Refuter has a winning strategy.*

Proof: Assume that $X \not\sim Y$ and Refuter uses the following strategy. In (b) he always chooses a pair (α', β') with the least eq-level, and in (c) he always chooses a move guaranteeing that $\text{EqLv}(\alpha', \beta') < \text{EqLv}(\alpha, \beta)$. Prop. 10 guarantees that this is possible and that $\text{EqLv}(\alpha', \beta') < \text{EqLv}(\alpha, \beta)$, or $\text{EqLv}(\alpha', \beta') = \text{EqLv}(\alpha, \beta)$ and $\text{SIZE}(\alpha', \beta') < \text{SIZE}(\alpha, \beta)$. Refuter thus must win eventually. \square

Lemma 12 (*Completeness.*) *If $X \sim Y$ then Prover has a strategy avoiding Refuter's win (the play might be infinite), on condition that the polynomial pol is sufficiently large.*

In the rest we show a proof of Lemma 12, by which a proof of Theorem 4 will be finished. From bisimulation games we borrow the terminology of a round-based game between **A** (Attacker, he) and **D** (Defender, she), played as follows: given a current pair (α, β) , **A** chooses some $\alpha \xrightarrow{r} \alpha'$ or $\beta \xrightarrow{r'} \beta'$; in the former case, **D** chooses some $\beta \xrightarrow{r'} \beta'$ where $\text{ACT}(r') = \text{ACT}(r)$, in the latter case **D** chooses some $\alpha \xrightarrow{r} \alpha'$ where $\text{ACT}(r) = \text{ACT}(r')$; the round is finished and (α', β') becomes the current pair for the next round. If a player is stuck then (s)he loses. It is obvious (by Prop. 10(2)) that if $\text{EqLv}(\alpha, \beta) = k < \omega$ then **A** has an *optimal strategy* (attaching a move to each pair (α_1, α_2)) guaranteeing his win within $k+1$ rounds. On the other hand, if $\text{EqLv}(\alpha, \beta) = \omega$ then **D** has a strategy keeping all current pairs in the play bisimilar (i.e., each current pair is an element of bisimulation equivalence). We now observe some simple facts (stated in convenient, not necessarily the strongest, forms).

We assume a given $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ and recall M, M_{rhs}, S_{rhs} from Def. 7.

Proposition 13 *Suppose $A\alpha \sim B\beta$ where $\|A\| \leq \|B\|$ and $\|A\| < \omega$.*

- (1) *There is γ such that $\alpha \sim \gamma\beta$ and $\text{SIZE}(\gamma) \leq M + M \cdot S_{rhs}$; we refer to such γ below.*
- (2) *If $\|B\| < \omega$ then either there is δ such that $\delta\alpha \sim \beta$, $\text{SIZE}(\delta) \leq M + M \cdot S_{rhs}$, and $\gamma\delta \neq \varepsilon$, in which case $\alpha \sim (\gamma\delta)^\omega$ and $\beta \sim (\delta\gamma)^\omega$, or γ satisfying (1) can be chosen so that $\|\gamma\| < \|B\|$.*
- (3) *If $A\gamma \not\sim B$ but $A\gamma\beta \sim B\beta$ then there is $\delta \neq \varepsilon$ such that $\beta \sim \delta\beta$, and thus $\beta \sim \delta^\omega$.*

Proof: (1) Given $(A\alpha, B\beta)$, suppose **A** plays a norm-reducing sequence $A\alpha \xrightarrow{u} \alpha$ ($|u| \leq M$) while **D** keeps bisimilarity, playing some $B \xrightarrow{u'} \gamma$ (i.e., $B\beta \xrightarrow{u'} \gamma\beta$); this yields the required $\alpha \sim \gamma\beta$ (where $\gamma\beta = \gamma$ when $\|\gamma\| = \omega$, by our convention after Def. 6).

(2) If there is no such δ (normed or unnormed) and **A** plays norm-reducing $B\beta \xrightarrow{u} \beta$, while **D** keeps bisimilarity, then α is exposed within $\|B\|$ moves; we get $\alpha \sim \gamma\beta$ where $\|\gamma\| < \|B\|$ (maybe $\gamma = \varepsilon$).

(3) If **A** uses an optimal strategy for $(A\gamma, B)$ from $(A\gamma\beta, B\beta)$ (ignoring the suffix β) while **D** keeps bisimilarity then the play obviously must reach a pair $(\beta, \delta\beta)$ or $(\delta\beta, \beta)$ where $\beta \sim \delta\beta$ and $\delta \neq \varepsilon$ ($\beta \sim \delta$ if $\|\delta\| = \omega$); we do not claim any bound on $\text{SIZE}(\delta)$ at the moment. \square

Hence if $\|A\| \leq \|B\|$, $\|A\| < \omega$, and $A\alpha \sim B\beta$, then one of the following points shows a bisimilar decomposition of $(A\alpha, B\beta)$ on condition that the size-conditions are satisfied:

1. (α, γ) , $(A\gamma, B)$, where $\text{SIZE}(\gamma) \leq M(1+S_{rhs})$, in the case $\|B\| = \omega$ and thus $\beta = \varepsilon$;
2. $(\alpha, (\gamma\delta)^\omega)$, $(\beta, (\delta\gamma)^\omega)$, $(A(\gamma\delta)^\omega, B(\delta\gamma)^\omega)$, where $\text{SIZE}(\gamma\delta)$ and $\text{SIZE}(\delta\gamma)$ are bounded by $2M(1+S_{rhs})$ (recall that, e.g., $(\gamma\delta)^\omega$ is, in fact, γ when $\|\gamma\| = \omega$);
3. $(\alpha, \gamma\beta)$, $(A\gamma, B)$, where $\|\gamma\| < \|B\| \leq M$ (in the case $A\gamma \sim B$);

4. $(\alpha, \gamma\delta^\omega), (\beta, \delta^\omega), (A\gamma\delta^\omega, B\delta^\omega)$, where $\|\gamma\| < \|B\| \leq M$ (in the case $A\gamma \not\sim B$).

From Prop. 8 we can derive that the size (and thus also the length) of γ and δ is bounded by an exponential function of the size of \mathcal{G} , except of δ in 4. But for this case an exponential bound is provided by Lemma 18 shown later: by putting there $\alpha_1 = A\gamma$, $\alpha_2 = B$ we get $\text{SIZE}(\delta) \leq (2M + |\mathcal{N}|^2 \cdot M_{rhs} + S_{rhs}) \cdot (1 + S_{rhs})$. It is thus clear that there is an exponential bound EB (in the size of \mathcal{G}) such that the size of each string in the pairs in 1., 2., 3., 4. is less than EB, when we except α , β , and $\gamma\beta$. To summarize:

Proposition 14 *If $A\alpha \sim B\beta$, $\text{SIZE}(A\alpha, B\beta) \geq \text{EB}$, and $\text{SIZE}(\alpha, \beta) < \text{SIZE}(A\alpha, B\beta)$ then $(A\alpha, B\beta)$ has a bisimilar decomposition, in one of the forms captured by 1., 2., 3., 4. above.*

We note in particular that if $\text{SIZE}(A\alpha, B\beta) \geq \text{EB}$ and $\text{SIZE}((A\alpha)_c) < \text{EB}$, $\text{SIZE}((B\beta)_c) < \text{EB}$ then $\text{SIZE}(\alpha, \beta) < \text{SIZE}(A\alpha, B\beta)$. (If the greater of $A\alpha, B\beta$ is just a cycle then $\text{SIZE}(\alpha, \beta) = \text{SIZE}(A\alpha, B\beta)$.)

To finish a proof of Lemma 12, let us imagine that Prover, starting from $X \sim Y$, only uses bisimilar decompositions of the type 1., 2., 3., or 4. in Point (b) of the Prover-Refuter game, whenever they are available. In (c) Prover always chooses so that she keeps $\alpha' \sim \beta'$; she thus maintains that each current pair is bisimilar (i.e., belongs to bisimulation equivalence). The next proposition helps to derive a bound on the work space which is sufficient for Prover to avoid losing by space-overflow. We refer to the canonical presentations of regular strings, and recall our convention after Def. 6.

Proposition 15 *If $\alpha \xrightarrow{a} \delta$ then $\delta_c \in \text{ROUND}(\alpha_c)$ or $\delta_c = \varepsilon$ and $\text{SIZE}(\delta_p) \leq S_{rhs} + \text{SIZE}(\alpha_p)$.*

Proof: $\alpha \xrightarrow{a} \delta$ implies $\alpha = A\alpha'$, $\delta = \gamma\alpha'$ for a rule $A \xrightarrow{a} \gamma$. If $\|\gamma\| = \omega$ then $\delta = \gamma$ and $\delta_c = \varepsilon$. If $\|\gamma\| < \omega$ then (also $\|A\| < \omega$ and) $\delta_c \in \text{ROUND}(\alpha_c)$ by Prop. 3(2). Moreover, δ_p is a prefix of $\gamma(\alpha')_p$ and $(\alpha')_p$ is a suffix of α_p (by Prop. 3(1)); hence $\text{SIZE}(\delta_p) \leq S_{rhs} + \text{SIZE}(\alpha_p)$. \square

Propositions 14 and 15 show that the above strategy of Prover guarantees that she maintains the following invariant for each current pair (α, β) (when $X \sim Y$):

1. $\alpha \sim \beta$;
 2. each of α, β is of the form $\gamma_1(\gamma_2)^\omega$ where $\text{SIZE}(\gamma_1) < \text{EB} + S_{rhs}$ and $\text{SIZE}(\gamma_2) < \text{EB}$.
- (1)

Hence there is indeed a polynomial pol such that the space $2^{pol(\text{size}(\mathcal{G}))}$ is sufficient for Prover to avoid losing (by space-overflow) when $X \sim Y$.

3.1 Exponential bounds on eq-levels in the normed case

Though we still assume a general $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$, we will describe, in fact, an optimal **A**'s strategy in the normed subcase, deriving the bound in Lemma 18 (which was used by our above proof of Theorem 4).

Remark. As already mentioned, the problem BPA-BISIM for normed BPA is known to be polynomial; the exponential bounds like in Lemma 18 seem to be derivable, e.g., from [3] but we provide a proof, to be self-contained.

We start with noting a technical fact. A path $\delta_0 \xrightarrow{r_1} \delta_1 \xrightarrow{r_2} \delta_2 \cdots \xrightarrow{r_k} \delta_k$ in $\mathcal{L}_{\mathcal{G}}^R$ is a *down-stair*, for the pair δ_0, δ_k , if $\|\delta_i\| > \|\delta_k\|$ for all $i, 0 \leq i \leq k-1$. (Hence all δ_j are normed, the norm can increase/decrease in single steps, but the least norm in the path is achieved precisely at the end.) Recall now the values M, M_{rhs} for our assumed $\mathcal{G} = (\mathcal{N}, \mathcal{A}, \mathcal{R})$ (Def. 7).

Proposition 16 *If $B \xrightarrow{u} \gamma$ is a shortest down-stair for B, γ then $|u| \leq |\mathcal{N}| \cdot M \cdot (1 + M_{rhs})$ and, moreover, each δ visited by the path $B \xrightarrow{u} \gamma$ satisfies $\|\delta\| \leq \|B\| + M_{rhs}$.*

Proof: Let $B \xrightarrow{u} \gamma$ be a shortest down-stair for the pair B, γ ($\|B\| > \|\gamma\|$). We can write $u = ru', r \in \mathcal{R}$, and we have either $B \xrightarrow{r} \delta\gamma \xrightarrow{u'} \gamma$, in which case ($\delta\gamma$ is an rhs and) $\delta \xrightarrow{u'} \varepsilon$ is a norm-reducing path, or $B \xrightarrow{r} \delta C\gamma_2 \xrightarrow{u'} \gamma = \gamma_1\gamma_2$, where $\|C\| > \|\gamma_1\|$; in the latter case we have $\delta C\gamma_2 \xrightarrow{u'_1} C\gamma_2 \xrightarrow{u'_2} \gamma_1\gamma_2$ where $u'_1 u'_2 = u'$, $\delta \xrightarrow{u'_1} \varepsilon$ is norm-reducing and $C \xrightarrow{u'_2} \gamma_1$ is a shortest down-stair for C, γ_1 (and thus $C\gamma_2 \xrightarrow{u'_2} \gamma_1\gamma_2$ is a shortest down-stair for $C\gamma_2, \gamma_1\gamma_2$).

Any shortest down-stair for a pair cannot contain a down-stair for the same pair; since $\|\gamma\| < \|B\|$, we have $|\gamma| < M$ and there are thus at most $|\mathcal{N}| \cdot M$ pairs C, γ_2 where $C \in \mathcal{N}$ and γ_2 is a suffix of γ . The claimed upper bounds are now obvious. \square

A (general) *strategy* of **A** attaches one move from α_k or β_k (if it exists) to each sequence $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_k, \beta_k)$ of (current) pairs in a play of the **A-D** game; such a strategy restricts the possible plays (to those where **A** only uses the prescribed moves).

Proposition 17 *If $\alpha \not\sim \beta$ then **A** has a strategy guaranteeing for any play starting from (α, β) that after a number of rounds the play reaches (α', β') such that one of the following holds: (1) $\|\alpha'\| \neq \|\beta'\|$, or (2) $\|\alpha'\| = \|\beta'\| < \omega$ and $\alpha' \not\sim_1 \beta'$, or (3) $\|\alpha'\| = \|\beta'\| = \omega$ and $\alpha' \not\sim \beta'$; moreover, $\text{SIZE}(\alpha', \beta') \leq \text{SIZE}(\alpha, \beta) + |\mathcal{N}|^2 \cdot M_{rhs} + S_{rhs}$.*

Proof: We assume $\alpha \not\sim \beta$ and consider an *optimal* strategy of **A**, ignoring the size-condition on (α', β') for the moment. Such a strategy obviously exists; from each current pair, which the strategy allows to reach when starting from (α, β) , it forces reaching (α', β') satisfying one of (1), (2), (3) in the least possible number of rounds.

We assume $\|\alpha\| = \|\beta\| < \omega$ (otherwise we are done), and consider a current pair $(A\delta_1, B\delta_2)$, where $\|A\delta_1\| = \|B\delta_2\| < \omega$; w.l.o.g. we assume $\|A\| \leq \|B\|$ and we put $\ell = \max\{\|\delta_1\|, \|\delta_2\|\} = \|\delta_1\|$. There are two disjoint possibilities:

- *either* we have a *top-remove* (of A), i.e., the strategy allows reaching a pair $(\delta_1, \gamma\delta_2)$ (after a number of rounds), where $\|\delta_1\| = \|\gamma\delta_2\|$ and there is a down-stair $B \xrightarrow{u} \gamma$,
- *or* we have a *top-win*, i.e., no possible continuation of the play contains **A**'s move $\alpha_1 \xrightarrow{r} \alpha_2$ such that $\|\alpha_2\| = \ell$ (and $\|\alpha_1\| = \ell+1$).

In the latter case we say that **A** keeps the play above ℓ , and we note that any future current pair $(A\delta'_1, B\delta'_2)$ (or $(B\delta'_2, A\delta'_1)$), where $\|A\delta'_1\| = \|B\delta'_2\|$, must constitute a top-remove case: otherwise **A** could force reaching the target from $(A\delta_1, B\delta_2)$ earlier if he played there as from $(A\delta'_1, B\delta'_2)$. (I.e., the move attached to a sequence (α, β) , $\text{SEQ}_1, (A\delta_1, B\delta_2)$, $\text{SEQ}_2, (A\delta'_1, B\delta'_2)$, SEQ_3 would be attached to the sequence (α, β) , $\text{SEQ}_1, (A\delta_1, B\delta_2)$, SEQ_3 .)

In any top-remove case $(A\delta_1, B\delta_2)$, with a corresponding $(\delta_1, \gamma\delta_2)$, we can safely change the strategy as follows: **A** performs (in a series of rounds) a chosen shortest down-stair $B \xrightarrow{u} \gamma$. During this series **D** is answering as she wants; either she keeps the equality of norms and

the play reaches the pair $(\delta_1, \gamma\delta_2)$, or she loses by reaching some (α', β') satisfying (1) or (2) where $\|\beta'\| \leq \|B\delta_2\| + M_{rhs}$ (by Prop. 16) and $\text{SIZE}(\alpha') \leq \max\{S_{rhs}, \|\beta'\| + M_{rhs}\}$.

Since a top-win case in a play can be encountered at most once for each unordered pair A, B , the (generous) bound $\text{SIZE}(\alpha', \beta') \leq \text{SIZE}(\alpha, \beta) + |\mathcal{N}|^2 \cdot M_{rhs} + S_{rhs}$ easily follows. \square

Lemma 18 *If $\alpha_1 \not\sim \alpha_2$ and $\alpha_1\beta \sim \alpha_2\beta$ then there is $\delta \neq \varepsilon$ such that $\beta \sim \delta\beta$ (and thus $\beta \sim \delta^\omega$, i.e. $\beta \sim \delta$ if $\|\delta\| = \omega$) and $\text{SIZE}(\delta) \leq (\text{SIZE}(\alpha_1, \alpha_2) + |\mathcal{N}|^2 \cdot M_{rhs} + S_{rhs}) \cdot (1 + S_{rhs})$.*

Proof: From $(\alpha_1\beta, \alpha_2\beta)$ we let **A** play a strategy from Prop. 17 for the initial pair (α_1, α_2) (ignoring the suffix β) and we let **D** play a strategy keeping bisimilarity. The play must obviously reach $(\alpha'_1\beta, \alpha'_2\beta)$ where $\alpha'_1\beta \sim \alpha'_2\beta$, $\|\alpha'_1\| \neq \|\alpha'_2\|$, and $\text{SIZE}(\alpha'_1, \alpha'_2) \leq b = \text{SIZE}(\alpha_1, \alpha_2) + |\mathcal{N}|^2 \cdot M_{rhs} + S_{rhs}$. From $(\alpha'_1\beta, \alpha'_2\beta)$ we let **A** play norm-reducing steps at the lesser-norm side; within b moves, the play reaches $\beta \sim \delta\beta$ where surely $\text{SIZE}(\delta) \leq b + b \cdot S_{rhs}$. \square

4 Additional Remarks

Since the content of the work-space cannot repeat when Refuter plays his optimal strategy in a case $X \not\sim Y$, Refuter wins within doubly exponential time; this also yields a doubly-exponential bound on the eq-level of nonbisimilar BPA processes.

All the pairs (α, β) which satisfy the invariant (1) (after Prop. 15) create a *basis* for \mathcal{G} , similar to the bisimulation base of [4] but with explicit regular strings. We could construct the basis by a standard coinductive approach (building a sequence of decreasing overapproximations). Each of the pairs in the basis fits into exponential space, and they are thus at most doubly-exponentially many of them.

Sénizergues [14] showed the decidability of bisimilarity for pushdown processes; BPA can be seen as single-state pushdown processes. It seems interesting to explore the decomposition approach here as well, using *regular terms* (as in [9]).

Finally we mention that closing the complexity gap between ExpTime and 2-ExpTime for bisimilarity on BPA is a natural future research topic; for weak bisimilarity on BPA even the decidability question is open.

References

- [1] J. Baeten, J. Bergstra, and J. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *J.ACM*, 40(3):653–682, 1993.
- [2] S. Böhm, S. Göller, and P. Jančár. Bisimilarity of one-counter processes is PSPACE-complete. In *CONCUR 2010 - Concurrency Theory*, volume 6269 of *LNCS*, pages 177–191. Springer-Verlag, 2010.
- [3] O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification on infinite structures. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, pages 545–623. North-Holland, 2001.
- [4] O. Burkart, D. Caucal, and B. Steffen. An elementary bisimulation decision procedure for arbitrary context-free processes. In *Proc. of MFCS'95*, volume 969 of *LNCS*, pages 423–433. Springer, 1995.

- [5] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. *Inf. Comput.*, 121(2):143–148, 1995.
- [6] W. Czerwiński and S. Lasota. Fast equivalence-checking for normed context-free processes. In *Proc. FSTTCS'10*, volume 8 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.
- [7] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theor. Comput. Sci.*, 158:143–159, May 1996.
- [8] P. Jančar. Strong bisimilarity on basic parallel processes is PSPACE-complete. In *Proc. LICS 2003*, pages 218–227. IEEE Computer Society, 2003.
- [9] P. Jančar. Decidability of DPDA language equivalence via first-order grammars. In *Proc. LICS 2012*. IEEE Computer Society, 2012.
- [10] M. Jurdzinski, J. Sproston, and F. Laroussinie. Model checking probabilistic timed automata with one or two clocks. *Logical Methods in Computer Science*, 4(3), 2008.
- [11] S. Kiefer. BPA bisimilarity is ExpTime-hard. *CoRR*, abs/1205.7041, 2012.
- [12] A. Kučera and R. Mayr. On the complexity of checking semantic equivalences between pushdown processes and finite-state processes. *Inf. Comput.*, 208(7):772–796, 2010.
- [13] R. Mayr. Weak bisimilarity and regularity of context-free processes is exptime-hard. *Theor. Comput. Sci.*, 330(3):553–575, 2005.
- [14] G. Sénizergues. The bisimulation problem for equational graphs of finite out-degree. *SIAM J. Comput.*, 34(5):1025–1106, 2005.
- [15] J. Srba. Strong bisimilarity of simple process algebras: complexity lower bounds. *Acta Inf.*, 39(6-7):469–499, 2003.
- [16] J. Srba. Beyond language equivalence on visibly pushdown automata. *Logical Methods in Computer Science*, 5(1), 2009.